P4-ENABLED SMART NIC: ARCHITECTURE AND TECHNOLOGY ENABLING SLICEABLE OPTICAL DCS

Yan Yan[1,2], Shen Tan[1], Arash Beldachi[2], Kalyani Rajkumar[2], Rui Wang[2], Reza Nejabati[2], Dimitra Simeonidou[2]

[1]*Raymax Technology Ltd., Hangzhou, China*
[2]*High Performance Networks Group, University of Bristol, Bristol, UK*
yan.yan@raymax.net

Keywords: P4, Smart NIC, Network Slicing

Abstract

P4-enabled Smart NIC is implemented and demonstrated. Interconnect with optical BVT, it offers agile 100Gbps interface to transport P4-defined data path for L2/L3 from VMs to sliceable optical transport. The Smart NIC can achieve 84.8Gbps throughput; its hardware-enabled SR can produce 30% more bandwidth.

1 Introduction

Large scale 5G deployments by major operators globally, along with advances in AR, VR, IoT, self-driving vehicles, have been promising enough to make network technologists and innovators to start the discussions what will be the next generation (i.e post 5G) of the network. SDN or centralised controller is being rolled out in the network or under the hood to enable orchestrated systems. NFV with virtualised network functions and elements with agile life cycle (that is on demand scaling, commissioning, re-purposing and reconfiguring, and relocating functions for mobility requirements) is going to be one of the main enablers for the next generation of the networking technologies.

Network Slicing across multiple layers, i.e optical, IP, and applications will be a critical feature in 5G networks and beyond. It is a virtualized technology framework that allows tailoring the network performance (latency, throughput) and functionality to the tenants' (mobile operator, DC applications, fintech, and so on) requirements. It enables forwarding the packets on the basis of the requests of the source, and dynamically demanding the bandwidth and latency.

In the IP domain, Segment Routing (SR) is one of the main candidates in providing virtualise layer2 and 3 VPNs. It utilises source routing and as an enhancement to MPLS, has been designed to utilise central controller for label assignment and distribution. [1] Various efforts have demonstrated source routing from servers to send traffic all the way through the core and finally to the destination. [2]

For optical networking, various vendors including the web scale internet companies, such as Facebook, have been developing compact optical transport systems [3] known as DCI, which are available in pizza box sizes such as IP/Eth solutions. They offer multi-rate multi-protocol client ports 10Gbps to 100Gbps, and variable bandwidth allocation BVT towards the core, where coherent technologies can take the signal for much longer distances.

1.1 DC architecture evolution and new open hardware initiatives

With DCs become flatter in architecture to accommodate east west traffic patterns, optical interconnects are getting closer to the network edge to offer higher bandwidth/cost efficiency, making servers with NICs becoming the network edge where policies, QoS will happen before traffic exists the server.

When communicating and controlling network equipment, the boxes which actually forward the packets and bytes in the network, essentially limit how the software can define them. There has been a number of attempts to introduce open hardware platforms [4]. Recently P4 [5] consortium has emerged as a result of research community efforts in providing a means to define packet processing pipelines on the fly. It has to be mentioned this consortium and the available standards and protocol are still in its infancy, and that makes efforts in utilizing this technology even more worthwhile so help shaping up its progress. Companies such as Barefoot networks are a proponent of bringing open standards to data plane. In addition, more recently major networking chip providers have

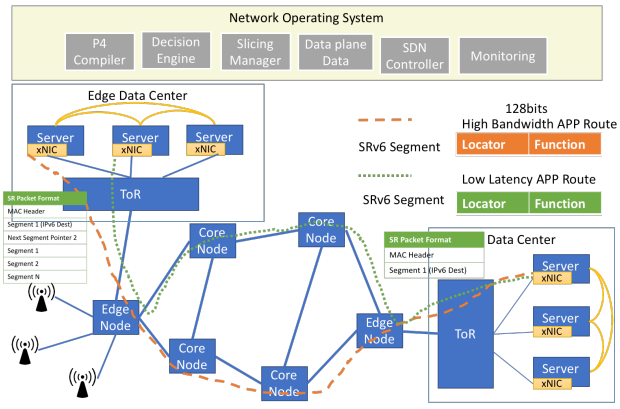


Fig. 1 Segment Routing Transportation in 5G network Edge Data Center to Core Data Center Architecture

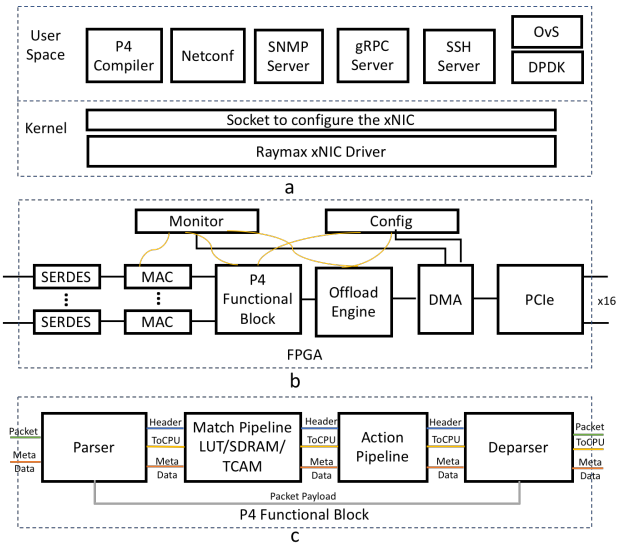


Fig.2 xNIC Architecture: (a) Software Architecture, (b) FPGA High Level Architecture, (c) P4 Functional Block Architecture

started to introduce P4 capabilities as the approach is gaining more traction.

1.2 FPGA unmatched capabilities

We chose the FPGA-based platform for our Smart NIC implementation. The most recent FPGA chipsets apart from field programmability and parallel processing capability, are equipped with high capacity network and storage IO and resources. This makes them extremely adaptive and responsive which can quickly respond to the network requirement and off-load the CPU. Xilinx puts lots of efforts on SDNet to enable the P4 to FPGA directly translation, however, for each P4 file, it takes hours to compile till FPGA.

2. Raymax Smart NIC enabled 5G sliceable inter-DC architecture

The proposed optical 5G inter-DC architecture in Fig.1, shows a converged fronthaul and backhaul network architecture, where virtualization and data plane programmability are key enablers. The P4-enabled Smart NICs, plugged into the servers, enable intra-rack server-to-server full mesh direct connection, which eliminate the electronics in the ToR, allowing a pure optical ToR (i.e SSS, WSS) [6] or a DCI to directly transmit over long distance. We were focusing on the edge DC to core DC end-to-end network slicing via segment routing. The Segment Routing (SR) IPv6 headers and MPLS labels can be inserted directly in the smart NIC by compiling the P4 files and downloaded to the Smart NIC through the socket. Compared to inserting the SR headers by the servers or the software, the P4 supplies better programmability and its enabled NIC offers better network performance and less CPU utilization. Fig.1 as an SRv6 example, displayed the way of by inserting diverse segment identifications, the packets can go through the routes by its segments only to a high bandwidth route (slashed) or a low latency route (dotted). When reaching to the end, the segments will be deleted by the end point server Smart NIC.

3. FPGA for P4-enabled Smart NIC architecture and design

The FPGA-based Smart NIC is capable of accelerating the network functions, like virtual switch, and also programmability, like P4. It has 2* 40Gbps or 2*100Gbps optical interfaces, and 12*25Gbps intra-rack connect interfaces.

The system design includes software part (Fig.2(a)) and FPGA-based data plane part (Fig.2(b)). We developed the kernel driver and DPDK driver to accommodate our FPGA-based Smart NIC. In addition to the kernel layer, we tried to use the open networking projects and existing protocols, i.e. P4C, ONOS and gRPC, shown in Fig.2(a), to provide data and control access to Smart NIC environment. In addition, we built our own P4 agent after the P4C to translate the json files to the binary files that FPGA can understand. The FPGA-based data plane part (Fig.2(b)) is implemented mainly for enabling the traffic flow between the SERDES and PCIe. The main functional block is P4 functional block and offload engine.

The implementation of the P4 data plane block (Fig 2(c)) in the FPGA was to follow the P4_16 language specification, and processed the packets as required. The P4 data plane functional block can be separated to parser, match, action and deparser. The packets processing was pipelined, while the packets were parsed, the metadata were extracted and matched with the mask table and matching table. Afterwards, the filtered packets followed the action rules, and then got repacked to a new packet for output. The parser was designed with two modes: one is the full parser mode that can parse the header of Ethernet, IPv4/IPv6, UDP, TCP, VxLAN, VLAN (3 nested), and MPLS (3 nested); the other one is the byte-based mode that can get the information from the P4 agent byte by byte on what to parse and how to parse.

The other major function blocks include the matching block and action block. For the matching block, considering the nature of P4 language and its converted binary mask file, to achieve the fast searching and matching, we employed TCAM to match the mask key and search key to search for the data in

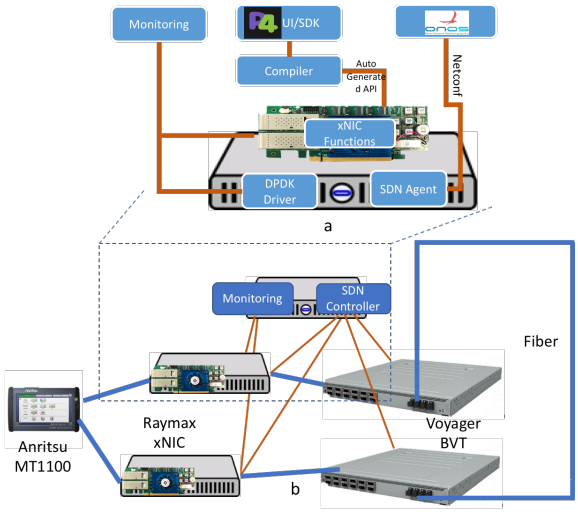
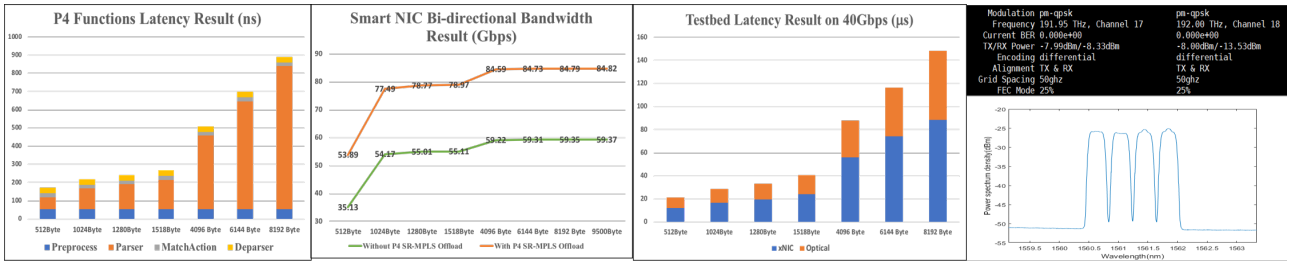


Fig.3 (a) Smart NIC setup, (b) Testbed setup



a b c d

Fig.4 (a)P4 functions Latency Result (b) Smart NIC bandwidth result (c) Testbed latency result (d) Voyager BVT setup and spectrum result.

the specific address with hit/miss indication. The implemented match block supported 10 to 660 mask keys and 32 matched output. Considering the match state '1', '0', 'X', it can support up to 3^660matching cases. The matching block can be cascaded for the use case with priority requirement of more than 32 output requirements.

For the action blocks, to realize the P4 standard set of primitive actions, we combined some similar actions, like a group of bit operation, grouped modify and set, and etc. to save the logic utilization. When the header was modified, the checksum was pipelined and calculated afterwards.

4. Results

4.1 Testbed setup

We setup a Smart NIC testbed with control plane and data Plane in Fig.3 (a). The Smart NIC was plugged into the server, and the server was installed with DPDK driver, SDN agent and ONOS controller. The P4 files could be pasted or uploaded to the SDK/UI, which would compile and translate the P4 files to the Smart NIC. There was a monitoring module for collecting and displaying the statics from Smart NICs and servers.

Furthermore, to emulate the edge-DC to DC environment, and displayed the IP domain segment routing combined with optical domain network slicing. We setup the optical testbed as displayed in Fig.3 (b) with Voyager BVT Transponders and back-to-back fibers. To measure the latency and bandwidth results, we employed Anritsu MT1100 traffic analyser to generate and analyse the result. The testbed was setup to assign different wavelengths to SR-MPLS labels, which enables server-to-server, end-to-end network slicing in both IP and optical domain.

4.2 P4 agility result

Our P4 compiler translates the .json file (compiled from .p4 file) to the P4 binary file that FPGA can understand. After the compiler, the .p4 file was translated to a mask table, a match table and an action table, which were sent to FPGA through PCIe. We measured the latency from click the P4 compile button till the smart NIC driver gets the translated binary file and is about to send to FPGA through PCIe. To avoid the system time inaccuracy, we wrote a script to run the tests 100 times and measured the whole latency. The latency result is **3.07s**, which proves the network manager can change the data plane behaviour in seconds. It is much lower compared to current Xilinx's P4-SDNet solution, which takes hours to complete the same functions.

4.3 Testbed Experimental Result

The measurement results were shown in Fig.4. We measured the P4 block latency in the FPGA (as in Fig.4(a)). The latency result revealed the detail latency of each functional block, as demonstrated in Fig.4 (a), the parser's latency was determined mainly by the Ethernet frame length, while other functional blocks' latency is comparable fixed.

The bandwidth result in Fig.4 (b) was measured with 1 CPU core in the PC of Intel Core i7-7700K CPU @ 4.20GHz x8, 62.8GiB Memory hardware setup. We inserted one SR-MPLS header to the packet and measured the maximum bandwidth of inserting by software (Without P4 SR-MPLS Offload) and inserting by FPGA (With P4 SR-MPLS Offload). The result demonstrated, with offload, the Smart NIC was able to achieve maximum 78.97Gbps throughput with 1518 Ethernet frame size and could go up to 84.82Gbps with jumbo frame size. Without offload, the bandwidth went down maximum 30%.

The whole testbed latency was planned to be measured by Anritsu MT1100, however, the MT1100 we used, only has one 100Gbps Ethernet port, but Voyager BVT needs 2*100Gbps ports to setup. Therefore, we measured the latency separately with MT1100 to NICs, then NICs to optical devices and fibers. The whole testbed latency in Fig.4(c) demonstrates in segments of the latency of smart NIC and optical devices.

Fig.4 (d) displayed optical results. We setup the Voyager BVT with QPSK modulation format. and the frequencies in 191.95THz, 192.00THz,192.05THz, 192.10THz, each of them matched to a MPLS segment enabling the optical domain slicing, and the spectrum result is in Fig.5 (b). It needs to be mentioned that, with the spec of Voyager BVT, with longer distance or higher bandwidth requirement, the modulation format can be tuned to 8QAM or 16QAM.

5 Conclusion

In this paper, we demonstrated the design, implementation and experimental result of P4-enabled smart NIC and its enabled inter-DC network slicing. In the experiment, with Voyager BVT, we were able to show the smart NIC's capabilities on enabling inter-DC end-to-end network slicing in both IP domain and optical domain. The measured results showed, with P4 SR-MPLS Smart NIC header insertion, the bandwidth performance can be up to 30% higher than without.

6 References

[1] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[2] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429-2453, thirdquarter 2018

[3] Guy Robers, "TIP Open Optical Packet Transport-A game-changer for R&E networking," in 9th CEF Networks Workshop, 2017

[4] 'Open Compute Project. OCP is reimagining hardware.' https://www.opencompute.org/ April 2019

[5] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford et al., "P4: Programming Protocol-independent Packet Processors," SIGCOMM Comput. Commun. Rev. 44, 3, July 2014

[6] Y. Yan et al., "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card," in Journal of Lightwave Technology, vol. 34, no. 8, pp. 1925-1932, 15 April15, 2016.